

Production Database Character Set Migration – Implementation Considerations

Thanks to Dwarak for sharing this document

Objective:

To support Multi-lingual characters in the Database and as the existing character set of the database, viz., US7ASCII does not support Multi-lingual characters. So the character set need to be converted to Unicode character set. Oracle offers two Unicode character sets, UTF8 and AL32UTF8.

Decision of choosing between UTF8 and AL32UTF8:

If the current database communicates (presently or in future) through database links with any database of version 8.0, it is better to opt for UTF8 character set. But otherwise, Oracle recommends to choose AL32UTF8 character set as AL32UTF8 is kept up-to-date with the Unicode standard and encodes the Unicode standards 3.0 (in database 9.0), 3.1 (database 9.2) or 3.2 (database 10g).

There are two ways to change the character set of a Database,

Approach 1

- Export the existing database, create a new Database with AL32UTF8 character set, and import the data from the export dump into newly created database.

Approach 2

- Alter the Character set of the existing DB to AL32UTF8 character set

Common steps to be followed irrespective of the above two approach

1. Run csscan tool on the current database
2. For all data marked as "lossy" check the currently stored data - it will be using codepoints that are invalid in the current db charset and they have to be adjusted.
3. For all data marked as "truncation" change the column sizes. This indicates that the data would not fit in AL32UTF8 database because it's using more bytes to store the same data in that character set.

Source – A.Kishore
<http://appsdba.info>

4. After all truncation and lossy data issues have been resolved run cscan again to verify that there now only is changeless and convertible data. If not then go back to steps 2 & 3.
5. Full Backup of the existing database
 - Using Export Utility
 - Hot backup

Approach 1

Implementation Considerations:

This is a safe approach as the existing production database is not affected by the character set Migration and there is no down-time.

Steps to be followed to create a new Database with AL32UTF8 character set

1. Follow the common steps
2. Create a new Database with AL32UTF8 Character set
3. Import from backup into the new AL32UTF8 database.
4. Run the cscan and check for lossy data.
5. Point the Application to this newly created database and check whether data is stored properly both for english and Multi-lingual.
6. Do Full Database Backup.

Approach 2

Implementation Considerations:

- By implementing this approach, *existing Multi-lingual data will not be changed and will not be displayed properly*; only data that are inserted in the future will be reflected properly. To overcome this, the Multi-lingual data need to be inserted again. Database Down-time is possible. But the down-time can be overcome by creating a clone of the existing database in a different host and this approach can be implemented, then the application can be pointed and tested.

Steps to be followed to Alter the Existing database to AL32UTF8 character set

1. Follow the common steps
2. connect as sys/password as sysdba
3. SHUTDOWN IMMEDIATE;
4. STARTUP MOUNT;
5. ALTER SYSTEM ENABLE RESTRICTED SESSION;

Source – *A.Kishore*

<http://appsdba.info>

6. ALTER SYSTEM SET JOB_QUEUE_PROCESSES=0;
7. ALTER SYSTEM SET AQ_TM_PROCESSES=0;
8. ALTER DATABASE OPEN;
9. ALTER DATABASE CHARACTER SET INTERNAL_USE AL32UTF8;
10. SHUTDOWN IMMEDIATE;
11. STARTUP;

Post Character Set Changes/Issues:

Changing varchar2 datatype to nvarchar2:

1. Wherever the Multi-lingual data is stored those columns need to be changed to Nvarchar2.

Reason for having nvarchar2 – As Multi-lingual characters will use more than one byte for each character the field length has to be increased to three times the current size, the same is not required in nvarchar2 datatype.